

Remarks

This RESPONSE is in reply to the Office Action mailed May 17, 2007. A Petition for Extension of Time is enclosed herewith, together with the appropriate fee. The fee for filing of a terminal disclaimer is also enclosed herewith. No fee is due for the addition of new claims.

I. Summary of Examiner's Rejections

Prior to the Office Action mailed May 17, 2007, Claims 1-30 were pending in the Application. In the Office Action, the Drawings and the Specification were objected to for minor informalities. Claims 1, 9, 11, 19, 21 and 29 were provisionally rejected on the grounds of non-statutory double patenting as being unpatentable over claims 1, 9, 11, 19, 21 and 29 of co-pending Application No. 10/777,362. Claims 10, 20 and 30 were rejected under 35 U.S.C. 112, second paragraph, as failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. Claims 11-19 were rejected under 35 U.S.C. 102(e) as being anticipated by Taylor et al. (U.S. Publication No. 2004/0019897, hereafter Taylor). Claims 1-10 and 20-30 were rejected under 35 U.S.C. 103(a) as being unpatentable over Taylor.

II. Summary of Applicant's Amendments

The present Response amends the Drawings, the Specification, and Claims 1, 11 and 21; cancels Claims 8, 10, 18, 20, 28 and 30; and adds new Claims 31-36, leaving for the Examiner's present consideration Claims 1-7, 9, 11-17, 19, 21-27, 29 and 31-36. A terminal disclaimer in compliance with 37 CFR 3.73(b) is also enclosed herewith, together with the appropriate fee. Reconsideration of the Application, as amended, is respectfully requested.

III. Objections to the Drawings

In the Office Action mailed May 17, 2007, the Drawings were objected to for minor informalities. Accordingly, enclosed is a Replacement Sheet for Figure 1 and Figure 2 that corrects the informalities. Applicant respectfully requests that the original drawing sheets for Figure 1 and Figure 2 be replaced with those of the Replacement Sheet. Reconsideration thereof is respectfully requested.

IV. Objections to the Specification

In the Office Action mailed May 17, 2007, the Specification was objected to for minor informalities. Accordingly, the Specification has been amended as shown above to correct the informalities. Reconsideration thereof is respectfully requested.

V. Double Patenting Rejections

In the Office Action mailed May 17, 2007, Claims 1, 9, 11, 19, 21 and 29 were provisionally rejected on the grounds of non-statutory double patenting as being unpatentable over Claims 1, 9, 11, 19, 21 and 29 of co-pending Application No. 10/777,362. Accordingly, enclosed herewith is an appropriate terminal disclaimer in compliance with 37 CFR 3.73(b), together with the appropriate fee. Applicant respectfully submits that this renders moot the rejection of these claims on the grounds of non-statutory double patenting, and reconsideration thereof is respectfully requested.

VI. Claim Rejections under 35 U.S.C. §112

In the Office Action mailed May 17, 2007, Claims 10, 20 and 30 were rejected under 35 U.S.C. 112, second paragraph, as failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. Accordingly, Claims 10, 20 and 30 have been canceled, rendering moot the rejection of these claims.

VII. Claim Rejections under 35 U.S.C. §102

In the Office Action mailed May 17, 2007, Claims 11-19 were rejected under 35 U.S.C. 102(e) as being anticipated by Taylor (U.S. Publication No. 2004/0019897).

Claim 11

Claim 11 has been amended to more clearly define the embodiment therein. As amended, Claim 11 currently defines:

11. *(Currently Amended) A method for loading software applications on a server, comprising the steps of:*

providing a software application, wherein said software application includes a plurality of modules and classes associated therewith;

parsing a control file associated with said software application, wherein said control file can be edited by a software developer and specifies a hierarchy of classloaders to be used with the modules in said software application, and wherein the hierarchy includes a plurality of branches that are specified by the software developer to provide namespace separation between different modules in the software application;

retrieving a selection of said classloaders according to the hierarchy specified by said control file; and

loading said modules and classes as part of said software application corresponding to said hierarchy, including, if a module in said software application is being redeployed then loading only the classloaders in the branches for that module, independently of other branches in the hierarchy.

Claim 11, as currently amended, defines that the control file can be edited by a software developer and specifies a hierarchy of classloaders to be used with the modules in said software application. The hierarchy includes a plurality of branches that are specified by the software developer to provide namespace separation between different modules in the software application. The method further comprises retrieving a selection of said classloaders according to the hierarchy, and loading said modules and classes as part of said software application corresponding to said hierarchy, including, if a module in said software application is being redeployed then loading only the classloaders in the branches for that module, independently of other branches in the hierarchy.

The advantages of the embodiment defined by Claim 11 include that it provides a more granular control over the classloader hierarchy structure and the associated interrelationships between classes of modules within a software application. This allows the developer to define their classloader organization according to their particular needs, for example placing modules frequently called by one another on the same branch (or in the same classloader), and therefore, in the same search path, while at the same time allowing the components to be optionally separated into different classloaders to facilitate dynamic reloading.

Taylor discloses a method, system, and program for processing objects in a distributed computing environment. A determination is made of a program is needed to process a component. A file including the determined program is requested from a remote process. The requested file is

received from the remote process, wherein the requested file includes a plurality of sections, each including different programs. At least one section includes programs that are intended only to be executed in a remote address space of the remote process and at least one other section includes programs that are intended to be downloaded from the remote process and execute in a client address space that is different than the remote address space. (Abstract). As further disclosed therein, the population manifest 326 is parsed by the container 306 to construct class loaders as necessary to load the components indicated in the manifests. FIG. 6 illustrates an example of a class loader hierarchy 350 that the population manifest 326 may implicitly define. The class loader hierarchy 350 divides class loaders into factory, root, and component class loaders. Component class loaders are used to load the variable components of the system, such as services, facility implementations, plug-ins, and dynamically loaded modules. Root class loaders load common interfaces that enable communication between the components of the system, such as facility interfaces. Factory class loaders are used to load factory classes. A factory is a component responsible for instantiating and destroying a particular type (or set of types) of components. A facility is a component whose instances can be shared across other components. (Paragraph [0050]). Each element in the one or more XML files forming the population manifest 326 indicates a component, encapsulated in a CAR file, to be loaded, and the attributes of the element may include information about the component, such as public package, version, etc. During runtime, a new component may be added to the system by adding a new element to the XML file comprising a population manifest 326. A root XML file may maintain information on the root components that are loaded, where the last XML element in the root XML file would comprise the component that is loaded by the lowest root class loader in the hierarchy. (Paragraph [0052]).

Applicant respectfully submits, as described by Taylor, the system therein is used to support programs that are intended only to be executed in a remote address space of the remote process, and at least one other section includes programs that are intended to be downloaded from the remote process and execute in a client address space that is different than the remote address space. As further described therein, the class loader hierarchy appears to use a manifest file at the application level. Similarly, Figures 7 and 8 appear to show that the class loader hierarchy stops at the population instance for the application, but does not appear to proceed to the module level within each population instance.

However, in the embodiment defined by Claim 11, as currently amended, the control file, which can be edited by a software developer, specifies a hierarchy of classloaders to be used with the modules (for example, the EJB and Web application files) in said software application. As further defined by Claim 11, the hierarchy includes a plurality of branches that are specified by the software developer to provide namespace separation between different modules in the software application. The method comprises retrieving a selection of said classloaders according to the hierarchy, and loading said modules and classes as part of said software application corresponding to said hierarchy, including, if a module, (such as an EJB or Web application file), is being redeployed then loading only the classloaders in the branches for that module, independently of other branches in the hierarchy, but not at the individual module level within an application. This allows for a more granular control over the classloader hierarchy structure and the associated interrelationships between modules within a software application. Applicant respectfully submits that Taylor does not appear to disclose or suggest such a feature.

In view of the above comments, Applicant respectfully submits that Claim 1, as amended, is neither anticipated by, nor obvious in view of, the cited references, and reconsideration thereof is respectfully requested.

Claims 12-19

Claim 18 has been canceled, rendering moot the rejection of this claim. Claims 12-17 and 19 depend from and include all of the features of Claim 11. Claims 12-17 and 19 are not addressed separately, but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and further in view of the amendments to the independent claims and the comments provided above. Reconsideration thereof is respectfully requested.

VIII. Claim Rejections under 35 U.S.C. §103

In the Office Action mailed May 17, 2007, Claims 1-10 and 20-30 were rejected under 35 U.S.C. 103(a) as being unpatentable over Taylor (U.S. Publication No. 2004/0019897).

Claims 1 and 21

The comments provided above with respect to Claim 11 are hereby incorporated by reference. Claims 1 and 21 have been similarly amended by the current Response to more clearly define the embodiments therein. For similar reasons as provided above with respect to Claim 11, Applicant respectfully submits that Claims 1 and 21, as amended, are likewise neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

Claims 2-10, 20 and 22-30

Claims 8, 10, 20, 28 and 30 have been canceled, rendering moot the rejection of these claims. Claims 2-7, 9, 22-27 and 29 depend from and include all of the features of either Claim 1, Claim 11 or Claim 21. Claims 2-7, 9, 22-27 and 29 are not addressed separately, but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and further in view of the amendments to the independent claims and the comments provided above. Reconsideration thereof is respectfully requested.

IX. Additional Amendments

Claims 31-36 have been newly added by the present Response. Applicant respectfully requests that new Claims 31-36 be included in the Application, and considered therewith.

X. Conclusion

In view of the above amendments and remarks, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.


Enclosed is a PETITION FOR EXTENSION OF TIME UNDER 37 C.F.R. §1.136 for extending the time to respond up to and including November 19, 2007.

Application No.: 10/777,361
Response to Office Action dated: May 17, 2007
Response dated: November 19, 2007

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: November 19, 2007

By: 
Karl F. Kenna
Reg. No. 45,445

Customer No.: 23910
FLIESLER MEYER LLP
650 California Street, Fourteenth Floor
San Francisco, California 94108
Telephone: (415) 362-3800

- 16 -

Attorney Docket No.: BEAS-01312US1 srm/kfk
kfk/wp/BEAS/1312/us1/1312us1.ROA_Reply051707.wpd